

Who is Mistaken?

Benjamin Eysenbach
MIT

bce@mit.edu

Carl Vondrick
MIT

vondrick@mit.edu

Antonio Torralba
MIT

torralba@csail.mit.edu

Abstract

Recognizing when people have false beliefs is crucial for understanding their actions. We introduce the novel problem of identifying when people in abstract scenes have incorrect beliefs. We present a dataset of scenes, each visually depicting an 8-frame story in which a character has a mistaken belief. We then create a representation of characters' beliefs for two tasks in human action understanding: predicting who is mistaken, and when they are mistaken. Experiments suggest that our method for identifying mistaken characters performs better on these tasks than simple baselines. Diagnostics on our model suggest it learns important cues for recognizing mistaken beliefs, such as gaze. We believe models of people's beliefs will have many applications in action understanding, robotics, and healthcare.

1. Introduction

In Figure 1, one person has a mistaken belief about their environment. Can you figure out who is mistaken? You likely can tell the woman is about to sit down because she incorrectly believes the chair is there. Although you can see the complete scene, the character inside the scene has an imperfect view of the world, causing an incorrect belief.

The ability to recognize when people have incorrect beliefs will enable several key applications in computer vision, such as in action understanding, robotics, and healthcare.

For example, understanding beliefs could prevent accidents by warning people who are unaware of danger, help robots to have more fluid interactions with humans [15], provide clues for anticipating human actions [14, 27], and better generate visual humor [6]. How do we give machines the capability to understand what a person believes?

In this paper, we take a step towards understanding visual beliefs. We introduce the novel problem of recognizing incorrect beliefs in short visual stories. We propose two new tasks aimed at understanding which people have false beliefs. Given a visual story, we aim to recognize **who** is mistaken and **when** they are mistaken. For example, in Figure 1, the woman is mistaken in the third frame.

To study this problem, we present a dataset of abstract scenes [34] that depict visual stories of people in various types of everyday situations. In each story, one or more people have mistaken beliefs, and we seek to recognize these people. Abstract scenes are ideal for studying this problem because we can economically create large datasets that focus on the high-level human activities, such as ones influenced by people's beliefs. The scenarios in our dataset are diverse and characters are mistaken for many reasons, such as occlusion or unexpected actions.

We present a simple model for learning to recognize mistaken characters in short sequences. Our model uses person-centric representations of scenes and combines information across several timesteps to better recognize mistaken characters. Experiments show that our model learns to recog-

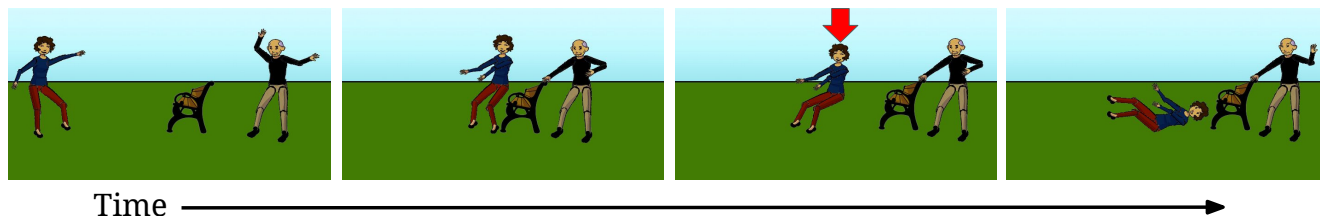


Figure 1: **Can you determine who believes something incorrectly in this scene?** In this paper, we study how to recognize when a person in a scene is mistaken. Above, the woman is mistaken about the chair being pulled away from her in the third frame, causing her to fall down. The **red arrow** indicates false belief. We introduce a new dataset of abstract scenes to study when people have false beliefs. We propose approaches to learn to recognize **who** is mistaken and **when** they are mistaken.



Figure 2: **Visual Beliefs Dataset:** We introduce a new dataset of abstract scenes to study visual beliefs. We show five example scenes from our dataset. The **red arrows** indicate that a person has a false belief in that frame. Each scene (row) contains eight images, depicting a visual story when read left to right. The caption below each scene was collected during annotation for visualization purposes only.

nize people’s beliefs better than baselines, suggesting that it is possible to make progress on understanding visual beliefs. Although we only train our model to predict mistaken beliefs, experiments suggest that our model is internally learning important cues for beliefs, such as human gaze.

The first contribution of this paper is introducing two new computer vision tasks for understanding people’s beliefs in short visual sequences of images. The second contribution is the release of a new dataset for both training and evaluating models for recognizing beliefs. The third contribution is a model for starting to tackle these belief tasks. In the remainder of this paper, we describe these contributions in detail. In Section 2, we review related work. In Section 3, we describe our new dataset and its collection procedure. In Section 4, we motivate the who and when tasks. In Section 5, we present several simple baseline models. In Section 6, we show experiments to analyze our models. We believe recognizing mistaken people in scenes can have a large impact in many applications, and we hope this paper will spur additional progress on this important problem. Code, models, data, and animated scenes will be available at <http://people.csail.mit.edu/bce/mistaken/>.

2. Related Work

Beliefs and Intentions: Our paper builds off several works that study the beliefs of people. In psychology, previous work has focused on theory of mind with people’s ability to reason about the beliefs of others. [24] note that peo-

ple use gaze-following for theory of mind, and failing to solve this problem may indicate a disability. From a computation perspective, [23] study theory-of-mind in robotics and its applications to human-robot interaction. [30] explore people’s intentions in real-world surveillance footage. [3] propose a Bayesian model for learning beliefs based on a partially-observable Markov Decision Process. [33] propose using probabilistic programming to infer the beliefs and desires of people in RGBD videos. Our approach similarly captures the beliefs of people, but we focus on how these beliefs differ from reality. We focus on learning the beliefs of characters directly from visual scenes.

Common Sense: Our work complements efforts to learn common sense. Understanding how people interact with each other and their world is an important step towards identifying their beliefs. [31] extract common sense from object detection corpora, while [8] learn visual common sense by browsing the Internet. [26] use abstract clip art images to learn how people, animals and objects are likely to interact. [17], [29] and [19] learn a model for physics given videos of colliding objects. [1] explore understanding social interactions in crowded spaces. [21] explore causality in unconstrained video to understand social games. In this work, we study the subset of common sense related to visual beliefs.

Activity Understanding: Our work is related to activity understanding in vision. [5, 28, 7, 20, 10]. Systems for understanding human actions typically leverage a variety of cues, such as context, pose, or gaze [22]. Our work complements action understanding in two ways. First, we study

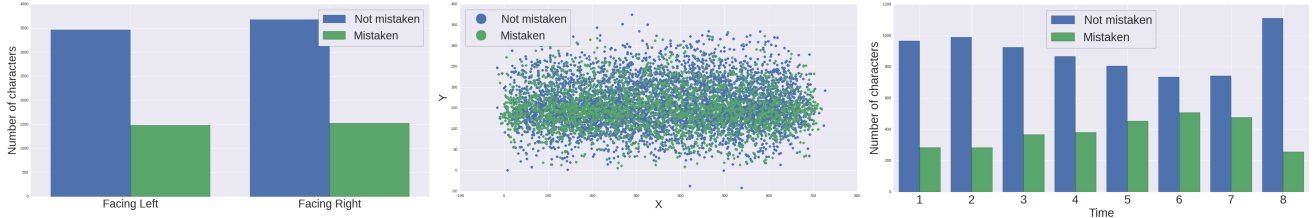


Figure 3: **Dataset Statistics:** We summarize statistics of our dataset. **(left)** Our dataset contains more not-mistaken characters than mistaken characters, but the ratio of mistaken/not-mistaken characters is the same for characters facing left and right. **(middle)** We show the (x, y) location of every character in every frame. The distribution for mistaken characters and not-mistaken characters appears similar. **(right)** There is a slight bias for people to be mistaken towards the end of the video. We compare (and outperform) against baselines that use this bias.

visual beliefs, which may be a useful signal for better understanding people’s activities. Second, recognizing visual beliefs often requires an understanding of people’s actions.

Abstract Images: We take advantage of abstract images pioneered by [34], which has become popular for studying high-level vision tasks. [2] use abstract images to learn features that encode the semantic similarity between images. [6] use abstract images to detect visual humor. [32] explore binary question-answering in abstract scenes, and [11] learn to predict object dynamics in clip art. While these approaches reason about image-level features and semantics, our approach looks at character-level features. Importantly, two characters in the same scene can have different beliefs about the world, so each character should have a different character-level feature. Additionally, we extend this previous work to multi-frame scenes depicting visual stories.

3. Dataset

We collected a dataset of abstract scenes to study the beliefs of characters. Each scene in our dataset consists of a sequence of 8 frames, and shows an everyday situation. In each scene, there is one or more people who believe something incorrectly about their environment. There are many reasons why a person may have a false belief, including occlusion and misinterpreting intentions. Although the characters inside the scenes do not know if they are mistaken, we designed the dataset so that third-party viewers can clearly recognize who is mistaken.

Our dataset complements existing abstract scene datasets. For example, our dataset builds upon the abstract scenes VQA dataset [2] in two ways: first, frames in our dataset are grouped into scenes telling stories over several timesteps; second, characters in our dataset frequently have mistaken beliefs. Additionally, our dataset complements the abstract humor dataset [6]. People with false beliefs may cause humor in a scene.

We believe abstract scenes provide a good benchmark for studying visual beliefs. We originally tried to collect a dataset of real videos containing people with false beliefs (such as suspense movies), but we encountered significant

difficulty scaling up dataset collection. While many real videos contain characters with mistaken beliefs, these beliefs are very complex. This complexity made large-scale annotation very expensive. We believe abstract scenes are suitable for understanding visual beliefs today because they allow the field to gradually scale up complexity on this important problem. In this work, we focus on mostly obvious false beliefs in abstract scenes.

We use our dataset for both learning and evaluation of models for detecting mistaken characters in scenes. We show a few examples of our dataset in Figure 2 and summarize statistics in Figure 3. We collected this dataset on Mechanical Turk [25]. First, we ask workers to illustrate scenes. Then, we ask workers to annotate mistaken characters. In the remainder of this section, we describe how we built this dataset. The appendix contains additional details.

3.1. Generating Scenes

In the illustration step, workers were told to drag and drop clipart people and objects into eight frames to tell a coherent story. The interface was a modified version of [2]. We told workers that some frame should contain a character who has a mistaken belief about the world. In addition to illustrating these eight frames, workers also wrote a scene-level description and eight frame-level descriptions. These descriptions were used during the annotation step, but were not used to train or evaluate our models.

The illustration interface was designed to ensure the scenes were diverse. First, the background of each scene was randomly chosen to be indoors or outdoors. Second, the people, animals, and objects an illustrator could choose to add were randomly chosen.

3.2. Annotation

In the annotation step, the goal was to label which characters have mistaken beliefs. We hired workers to review the previously illustrated scenes, and write one yes/no question for each frame. For each frame, workers wrote the true answer to the question and the answer according to each

of the characters. We labeled a character as mistaken if its answer was different from the true answer.

In total, we collected 1,496 scenes, 1,213 of which passed our qualification standards. These scenes were the collective effort of 215 workers. On average, each frame contains 1.71 characters, and an average of 0.53 characters are mistaken per frame; characters are mistaken in 31.03% of frames. A pool of 237 workers annotated each scene twice. The labels for whether a character was mistaken were consistent between workers 61.15% of the time, indicating that in some scenes it was unclear whether a character was mistaken. In this paper, we only consider scenes where characters are clearly mistaken or not.

3.3. Quality Control

In pilot experiments, we found that many workers misunderstood the task. We used three tools to filter out bad scenes and teach workers how to create good scenes. First, workers were required to complete a qualification quiz before starting the illustration and annotation steps. In the quiz for the illustration task, workers identified good and bad scenes. In the quiz for the annotation step, workers filled in characters’ answers for a scene with preselected questions. These quizzes forced workers to think about the beliefs of characters. Adding these quizzes significantly increased the quality of our data. Second, for the illustration task we manually reviewed the first scene completed by each worker. If the scene was incoherent or did not contain a mistaken character, we disallowed the worker from illustrating more scenes. Third, we showed annotators the scene-level and frame-level descriptions for the scenes they were illustrating, helping them understand what the illustrator intended.

3.4. What Causes Mistaken Beliefs?

What causes characters to have mistaken beliefs? Figure 2 shows a few scenes from our dataset that highlight different types of mistaken beliefs. In the first scene, the woman is mistaken because the dog is **occluded** behind couch, and because she cannot see actions **outside her field of view**. In the second scene, the woman falsely accuses the boy of breaking the painting because she cannot observe events when she is **not present**. The girl in the third scene mistakenly assumes the boy can safely get off the teeter totter because of her **faulty reasoning about physics**. In the fourth scene, the boy wearing a red shirt **misinterprets the intentions** of the other boy. In the last scene, the woman wearing the red shirt lacks the **common sense** that some mushrooms are poisonous. Recognizing mistaken characters requires detecting each of these types of mistaken beliefs.

To recognize characters with these types of incorrect beliefs, models will need a rich understanding of the scene, such as action recognition, gaze following, physical reason-

ing, and common sense knowledge. In this work, we take the first step and tackle a subset of these challenges.

4. Belief Tasks

We study two tasks for recognizing mistaken people:

Task 1: Who is mistaken? In this task, given a scene and a character, the goal is to predict whether the character is mistaken in any frame in the scene. This task has several applications in identifying people who may be confused or unaware of danger.

Task 2: When are they mistaken? In this task, given a frame, the goal is to predict whether any character is mistaken in this frame. This task has applications in identifying when people might be confused, but it is not possible to know who is confused, such as in a crowd.

Joint Task: We also explore a joint task where we seek to simultaneously recognize who is mistaken as well as localize when they are mistaken in time.

5. Method

We now describe an approach for predicting who is mistaken and when they are mistaken. Recognizing mistaken characters requires looking beyond a single frame as knowledge of the past or knowledge of the future can provide important signals for recognizing mistakes. For example, in the second scene of Figure 2, a model must see that the woman was not present when the girl broke the painting to understand why she falsely accused the boy. Our model for detecting mistaken characters will look at the past, present, and future. The model must also understand what a person may know and what they might not. To understand that a person is mistaken, the model should be able to determine that the scene is different from what the person believes.

5.1. Person-Centric Representation

Before predicting whether a character is mistaken, we must tell our model which character to focus on. We use a **person-centric** representation of the world, where the model takes the perspective of an outside observer focusing on a specific character. For each frame in the scene, we center the frame at the head of the specified character. We also flip the frame so the specified character always faces left. For example, in Figure 4, the frame in the upper left can be viewed from each of the three characters’ perspectives. Alternative approaches that remove parts of the frame outside the character’s field of view may struggle to reason about what the character cannot see.

5.2. Visual Features

We use a frame-wise approach by extracting visual features for each frame and concatenating them temporally to create a time-series. We extract visual features from the

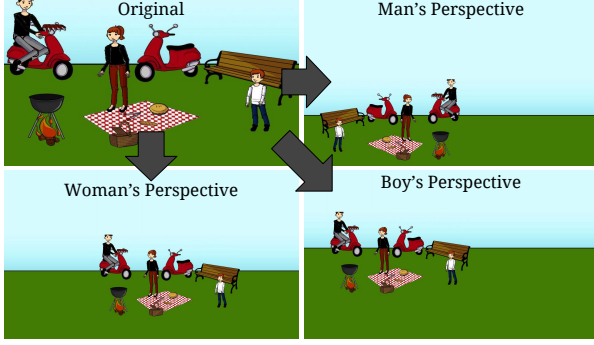


Figure 4: **Person-Centric Representation:** We use a visual representation that focuses on the character of interest.

person-centric images using the AlexNet convolutional network [16] trained on ImageNet [9]. We use activations from POOL5, and further downsample by a factor of two. The resulting feature has size (256, 12, 21).

One alternative is to use a handcrafted representation which exploits the fact that we have parameters for the rendering model of our abstract scenes. Using natural image features may allow for easier domain adaptation to real images. Moreover, although the features we use are trained on natural images (i.e. ImageNet), we found success at using them in abstract scenes, possibly because the quality of abstract scenes is high enough.

5.3. Learning

To learn to predict whether a person is mistaken or not, we can train a regularized convolutional logistic regression model, supervised by annotations from our training set. Suppose our image sequences are length T and our features are D dimensional. Let $\phi(x_i, p_j) \in \mathbb{R}^{T \times D}$ represent the features for sequence x_i for person p_j and $y_{i,j} \in \{0, 1\}^T$ be our target category binary, indicating whether person p_j is mistaken in each frame of sequence x_i . Our vector of predictions is $\hat{y}_{i,j} \in \mathbb{R}^T$. We optimize the objective:

$$\min_w \sum_{i,j,t} (y_{i,j}^t \log(\hat{y}_{i,j}^t) + (1 - y_{i,j}^t) \log(1 - \hat{y}_{i,j}^t)) \quad (1)$$

where $\hat{y}_{i,j}^t = (w * \phi(x_i, p_j) + b)^t$

The learned weight vector $w \in \mathbb{R}^{K \times D}$ represents the convolutional kernel, where parameter K specifies the temporal width; $b \in \mathbb{R}$ is the learned bias. For simplicity, we omit the L2 penalty on w . The superscript $(\cdot)^t$ gives the entry of a vector corresponding to frame t in a scene. We denote convolution as $*$, which is performed temporally. To handle border effects, we pad these features with zeros. The convolutional structure of our model encodes our prior that character’s beliefs are temporally invariant.

Method	Task		
	Who+When	Who	When
Chance	50	50	50
Time	62.9 (1.9)	52.4 (1.8)	64.3 (2.2)
Pose	51.9 (2.1)	50.3 (3.5)	54.8 (1.9)
Time+Pose	60.6 (2.0)	51.6 (1.2)	61.2 (1.9)
Single Image	61.1 (1.7)	59.7 (3.3)	62.0 (2.0)
Multiple Image	66.6 (1.8)	64.1 (2.8)	67.5 (1.8)

Table 1: **Quantitative Evaluation:** We evaluate the accuracy of our model versus various baseline on the who task, the when task, and the joint task. We report classification accuracy; parenthesis show standard deviations.

5.4. Who and When

We tackle two tasks related to beliefs. We seek to predict who is mistaken, and when they are mistaken. We train a single model that can be used for both tasks. Given a sequence x_i^t centered at time t and a person p_j in the sequence, we train a model to estimate whether person p_j is mistaken at time t . To answer the who question, one can then marginalize the classifier response across time. Likewise, to answer the when question, one can marginalize the classifier response across people.

5.5. Implementation Details

We extracted image features using Caffe [12] and we used Keras with Theano [4] for learning. To optimize the weights, we used Adam [13], with a learning rate 10^{-5} and a batch size of 32. We added weight decay with parameter 1, and stopping training after the validation accuracy had stopped increasing for 3 consecutive iterations. We found weight decay helped prevent overfitting.

6. Experiments

We analyze several models on our dataset of abstract scenes. We evaluate each model on the “who” task, the “when” task, and the joint “who + when” task.

6.1. Experimental Setup

We trained each model on the joint task: given a character and a frame, classify if this character is mistaken in this frame. Before training, we balance the dataset by resampling so 50% of training examples have a mistaken character. We randomly divide the dataset into training/validation/testing splits with sizes 80%/10%/10%. For the experiments in Table 1, we repeat each experiment 20 times with different splits, and report the mean and standard deviation of the accuracies. For the numbers in Table 2, we only repeat each experiment 6 times due to cost.

6.2. Baselines

Our dataset contains some biases surrounding mistaken characters. For example, particular characters in certain

Method	Task		
	Who+When	Who	When
Chance	50	50	50
Multiple Image	66.6 (1.8)	64.1 (2.8)	67.5 (1.8)
Flipped	54.5 (1.8)	52.5 (1.7)	55.8 (2.4)
Centered	62.4 (2.5)	55.6 (3.0)	63.0 (2.4)
Rewind	57.4 (2.8)	61.4 (3.6)	57.3 (1.8)

Table 2: **Ablation Analysis:** We study the impact of training on altered data and testing on unaltered data. During training, we modify data to flip the character’s pose (Flipped), not use the person-centric representation (Centered), and show the frames in reverse order (Rewind). The decrease in accuracy on each task indicates that pose, the person-centric representation, and the arrow of time are important parts of our model.

scenes are more likely to be mistaken than others. To explore these low-level biases, we use three simple features, and fit a kernelized SVM (RBF kernel) to each:

Time: This model uses only the temporal position of the frame within the scene, represented as a fraction between 0 and 1. This model can exploit the bias that frames in the middle of the scene are more likely to contain a mistaken character than frames at the beginning or end.

Pose: This model uses only the pose of the indicated character. Pose includes the (x, y) position of the character, as well as a boolean indicator of whether the character is looking left or right. The (x, y) coordinates are normalized to be in the interval $[0, 1]$.

Time + Pose: This model combines the features from the Time model and the Pose model.

Single Image: This model exploits biases that may be present in a single image. For example, characters may be more likely to be mistaken if they are looking away from the other characters in the frame. We trained a Single Image logistic regression model on person-centric visual features extracted from only the frame of interest.

6.3. Who is mistaken?

In this experiment, each model is given a scene and a character, and must determine whether the character is mistaken in any frame. The (scene, character) pairs are randomly sampled so 50% of examples contain a mistaken character. We evaluate the model’s decision function on each frame in the scene. For the SVM-based baseline models, each prediction is the signed distance from the separating hyperplane; for the image models, which use logistic regression, each prediction is a value in the interval $(0, 1)$. We take the maximum of these frame-level predictions as the model’s scene-level prediction. To obtain a binary decision, we threshold this scene-level prediction (at 0 for the SVM models, and at 0.5 for the logistic regression models).

The second column of Figure 1 shows that our Multiple

Image model achieves a higher accuracy on the “who” task than the four baselines. The baselines using time and poses perform slightly above chance. This indicates that position of a frame in a scene and the position of a character within a frame are not good indicators of whether a character is mistaken. The Single Image model does better than these simple baselines, but worse than the Multiple Image model. This suggests that individual frames contain some information about which character’s are mistaken. It also suggests that our model is using information across frames to better predict which character is mistaken.

6.4. When are they mistaken?

In this experiment, each model predicts whether any character in a frame is mistaken. Frames are randomly sampled so exactly 50% contain mistaken characters. We evaluate the model’s decision function on each character’s person-centric representation of the scene. As in the “who” experiment, we aggregate predictions across characters by taking the maximum of the model’s decision function.

The third column of Table 1 shows that our model achieves a higher accuracy on the “who” task than the four baselines. The Time baseline achieves a fairly high accuracy, indicating that the position of a frame within a scene is a good indication of whether the frame contains a mistaken character. The Pose baseline is also above chance, suggesting that a character’s position and orientation within a frame are correlated with whether the character is mistaken. The Single Image model performs better than the Pose model, suggesting there is information in the images besides pose that is important for recognizing mistaken beliefs. This might include the character’s relative position to people and objects in the scene. Our Multiple Image model performs better than the Single Image model, suggesting that our model is using information across frames.

6.5. Joint Task: Who and When?

In this experiment, the goal is to predict whether a character is mistaken in a given frame. Frames are randomly sampled so 50% of (frame, character) pairs contain a mistaken character. As shown in the first column of Table 1, our model achieves a higher accuracy on the “who” task than the four baselines. Similar to the “when” experiment in Section 6.4, the Time baseline achieves a fairly high accuracy on this joint task. The Pose baseline performs poorly, suggesting that the Time+Pose model likely ignores pose. Although pose is a poor feature for the “who + when” task, other features of a single image are important: the Single Image model performs well without knowing the position of the frame in the sequence. However, the Multiple Image model performs better than the Single Image model by aggregating features across time.

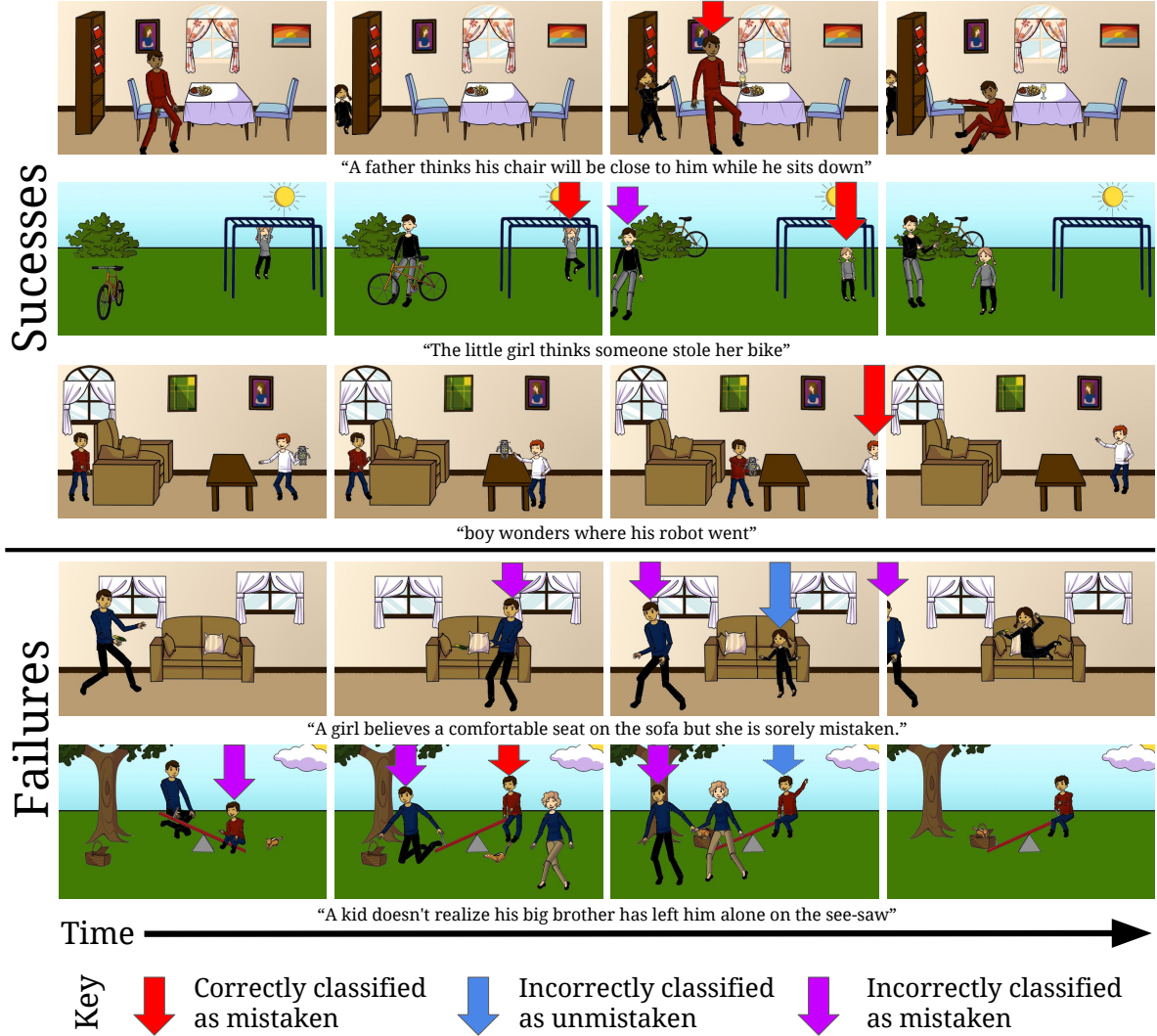


Figure 5: **Example Results:** We show predictions from our model. The first three rows show correct predictions. Our model fails to detect mistaken characters in the last two scenes, which require reasoning about occlusion and physics.

6.6. Qualitative Results

Figure 5 shows our model’s predictions on five scenes. **First Row:** Our model correctly detects that the man is mistaken in the third frame when the girl is about to pull his chair from beneath him. In this scene, the man is mistaken because he cannot see the girl’s actions behind him. **Second Row:** Our model correctly predicts that the girl is mistaken in the second and third frames as she can not see the man take her bike. Our model incorrectly predicts that the man is also mistaken in the third frame. This is possibly because our model has learned that a character is likely to be mistaken when another character is performing actions behind it. **Third Row:** Our model correctly identifies the boy wearing a white shirt as mistaken in the third frame. **Fourth Row:** The man plays a prank on the girl by hiding a piece of corn beneath a pillow. Our model incorrectly pre-

dicts that the man is mistaken, likely because he cannot see the actions of the girl behind him. Our model incorrectly predicts that the girl is not mistaken in the third frame, perhaps because the corn is occluded behind the pillow. Our model might think that the corn disappeared when it became occluded. **Fifth Row:** We show another failure case in which a man places a basket on the see-saw, leaving the boy stranded. Here, our model incorrectly predicts that the boy has a misbelief in the first frame, but does not have a misbelief in the third frame. Understanding this situation requires knowledge of basic physics, which our model currently lacks. Advances in physical understanding may improve reasoning about visual beliefs. Our model thinks the man playing the prank is mistaken, but predicts that the child being pranked is not mistaken. We believe models for visual humor could improve our results.



Figure 6: **Predictions from Ablation Experiments:** We visualize our ablation experiments. The first and third rows show a normal scene, and the second and fourth rows show perturbed scenes. **Row 1:** The first row shows a normal scene and predictions from our model. **Row 2:** In the second row, we flip the boy’s pose. Notice in the last frame, the boy no longer sees the girl so our model predicts he is still mistaken. **Row 3:** The third row shows another normal scene. **Row 4:** Predictions from the Rewind model make sense for the frames in the fourth row: the woman is mistaken in the second and third frames because she does not see the dog put the pie on the table, and therefore does not know how the pie appeared.

6.7. Effect of Person-centric Representation

We tested the impact of our person-centric representation by training a **Centered** version of our Multiple Image model without using the person-centric representation for each character. To avoid decreasing the image feature dimension, we created a new image with the original frame in the center. Table 2 shows the performance of this Centered model. Lacking a representation of the character of interest, this model performs worse on the “who” task compared to our model. For the “when” task, which asks whether any character is mistaken, this model performed only slightly worse than our model. For scenes containing a single character, detecting whether any character is mistaken is similar as detecting whether a specific character is mistaken. It is plausible that the Centered model beat chance because it made correct predictions on frames containing a single character. However, our results suggest that the person-centric representation helps recognize visual beliefs.

6.8. Effect of Gaze

Next, we evaluated the effect of gaze on our model. We trained a **Flipped** variation on our Multiple Image model

that flipped the character’s pose during training. In the person-centric representation, the character of interest now looked right instead of left. We also removed the character of interest from the frame to avoid creating unrealistic images. For example, if we flipped a character sitting on a chair, his limbs would now extend through the back of the chair. During testing, we used the unmodified person-centric representation.

If our model ignored the direction the character was facing Table 2 shows that this Flipped model performs worse than our model on the three tasks. If our model ignored the direction the character was facing, flipping the character would have little effect on the model performance. Flipping the character had a noticeable effect, suggesting that our model has learned that characters’ gaze is important for recognizing mistaken characters. Moreover, this suggests the model is starting to follow gaze [22] without supervising it to do so. In Figure 6, the top two rows compare predictions made by our original model and the Flipped variation.

6.9. Effect of Arrow of Time

Table 1 shows that our Multiple Image models recognizes mistaken characters more accurately than the Single

Image model. To study how our model aggregates temporal information across frames, we trained a **Rewind** variation. The Rewind model watches each scene backwards during training, but forwards during testing. If our model ignored time, the Rewind model would perform similarly. If our model pooled features over time, but ignored in which frames certain features were present, the Rewind model would perform the same.

Table 2 shows that the Rewind model performs worse than our model. This suggests that our model treats past and future frames differently, and that the arrow of time [18] is important. The accuracy of the Rewind model is most similar to the accuracy of our model on the “who” task, a task which does not require reasoning about time. In Figure 6, the bottom two rows compares predictions made by our original model and the Rewind variation.

7. Discussion

We propose a new computer vision task to recognize when people have mistaken beliefs about their environment. We introduce a dataset of abstract scenes to study this problem. Abstract scenes are attractive for making progress on this problem because they enable us to gradually scale up complexity. Our dataset consists of diverse situations where people can be mistaken for various reasons, such as occlusion, intention, or surprise. We present a simple model that uses multiple timesteps and a person-centric representation of the scene. Although we only supervise the model with indicators of which characters are mistaken, our ablation experiments suggest that the model learns important cues for this task, such as gaze. We believe understanding people’s beliefs can enable many applications in action understanding, healthcare, and robotics.

Acknowledgements: We thank workers on Mechanical Turk for their creative scenes. NVidia donated the GPUs used for this research. This work was supported by a Samsung grant to AT, a Google PhD fellowship to CV, and MIT UROP funding to BE.

References

- [1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2
- [2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. VQA: Visual Question Answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015. 3
- [3] C. L. Baker, R. R. Saxe, and J. B. Tenenbaum. Bayesian theory of mind: Modeling joint belief-desire attribution. In *Proceedings of the thirty-second annual conference of the cognitive science society*, pages 2469–2474, 2011. 2
- [4] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*, 2012. 5
- [5] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Nieves. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015. 2
- [6] A. Chandrasekaran, A. Kalyan, S. Antol, M. Bansal, D. Batra, C. L. Zitnick, and D. Parikh. We Are Humor Beings: Understanding and Predicting Visual Humor. *arXiv preprint arXiv:1512.04407*, 2015. 1, 3
- [7] Y.-W. Chao, Z. Wang, Y. He, J. Wang, and J. Deng. HICO: A benchmark for recognizing human-object interactions in images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1017–1025, 2015. 2
- [8] X. Chen, A. Shrivastava, and A. Gupta. Neil: Extracting visual knowledge from web data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1409–1416, 2013. 2
- [9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 5
- [10] A. Fathi, Y. Li, and J. M. Rehg. Learning to recognize daily actions using gaze. In *European Conference on Computer Vision*, pages 314–327. Springer, 2012. 2
- [11] D. F. Fouhey and C. L. Zitnick. Predicting object dynamics in scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2019–2026, 2014. 3
- [12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014. 5
- [13] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5
- [14] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert. Activity forecasting. In *European Conference on Computer Vision*, pages 201–214. Springer, 2012. 1
- [15] H. S. Koppula and A. Saxena. Learning Spatio-Temporal Structure from RGB-D Videos for Human Activity Detection and Anticipation. In *ICML (3)*, pages 792–800, 2013. 1
- [16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 5
- [17] A. Lerer, S. Gross, and R. Fergus. Learning Physical Intuition of Block Towers by Example. *arXiv preprint arXiv:1603.01312*, 2016. 2
- [18] L. C. Pickup, Z. Pan, D. Wei, Y. Shih, C. Zhang, A. Zisserman, B. Scholkopf, and W. T. Freeman. Seeing the arrow of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2035–2042, 2014. 9
- [19] L. Pinto, D. Gandhi, Y. Han, Y.-L. Park, and A. Gupta. The Curious Robot: Learning Visual Representations via Physical Interactions. *arXiv preprint arXiv:1604.01360*, 2016. 2

- [20] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2847–2854. IEEE, 2012. 2
- [21] K. Prabhakar, S. Oh, P. Wang, G. D. Abowd, and J. M. Rehg. Temporal causality for the analysis of visual events. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1967–1974. IEEE, 2010. 2
- [22] A. Recasens, A. Khosla, C. Vondrick, and A. Torralba. Where are they looking? In *Advances in Neural Information Processing Systems*, pages 199–207, 2015. 2, 8
- [23] B. Scassellati. Theory of mind for a humanoid robot. *Autonomous Robots*, 12(1):13–24, 2002. 2
- [24] S. V. Shepherd. Following gaze: gaze-following behavior as a window into social cognition. *Frontiers in integrative neuroscience*, 4:5, 2010. 2
- [25] A. Sorokin and D. Forsyth. Utility data annotation with amazon mechanical turk. *Urbana*, 51(61):820, 2008. 3
- [26] R. Vedantam, X. Lin, T. Batra, C. Lawrence Zitnick, and D. Parikh. Learning common sense through visual abstraction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2542–2550, 2015. 2
- [27] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating the future by watching unlabeled video. *arXiv preprint arXiv:1504.08023*, 2015. 1
- [28] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176. IEEE, 2011. 2
- [29] J. Wu, I. Yildirim, J. J. Lim, B. Freeman, and J. Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In *Advances in Neural Information Processing Systems*, pages 127–135, 2015. 2
- [30] D. Xie, S. Todorovic, and S.-C. Zhu. Inferring Dark Matter and Dark Energy from Videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2224–2231, 2013. 2
- [31] M. Yatskar, V. Ordonez, and A. Farhadi. Stating the Obvious: Extracting Visual Common Sense Knowledge. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 193–198, San Diego, California, June 2016. Association for Computational Linguistics. 2
- [32] P. Zhang, Y. Goyal, D. Summers-Stay, D. Batra, and D. Parikh. Yin and Yang: Balancing and answering binary visual questions. *arXiv preprint arXiv:1511.05099*, 2015. 3
- [33] Y. Zhao, S. Holtzen, T. Gao, and S.-C. Zhu. Represent and Infer Human Theory of Mind for Human-Robot Interaction. In *2015 AAAI Fall Symposium Series*, 2015. 2
- [34] C. L. Zitnick and D. Parikh. Bringing semantics into focus using visual abstraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3009–3016, 2013. 1, 3

Appendix

In these supplemental materials, we discuss the dataset in more depth. In Section A, we discuss how workers illustrated scenes for our dataset. Section B describe how workers annotated scenes. In Section C, we animate our scenes.

A. Illustrating the Dataset

In this section, we describe how workers illustrated scenes in our dataset.

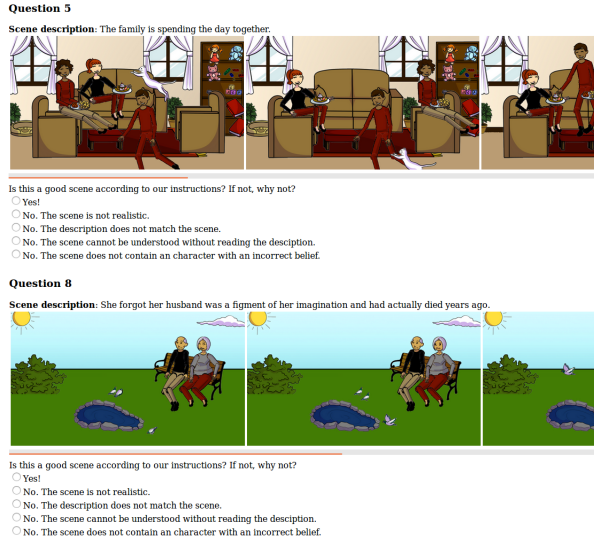


Figure 7: **Illustration quiz:** Two scenes from the illustration quiz. Workers scrolled left/right to see the 8 frames in each scene.

A.1. Illustration Quiz

The first time workers logged in, they were presented with a quality control quiz. In this quiz, workers were shown a number of scenes, and were asked “Is this a good scene according to our instructions? If not, why not?” Workers chose one of the following options:

- Yes!
- No. The scene is not realistic.
- No. The description does not match the scene.
- No. The scene cannot be understood without reading the description.
- No. The scene does not contain a character with an incorrect belief.

Figure 7 shows some of the scenes from our quiz. Note that the workers could scroll left/right to see all eight frames in the scene. Workers could begin illustrating their own scenes only after correctly completing the quiz.

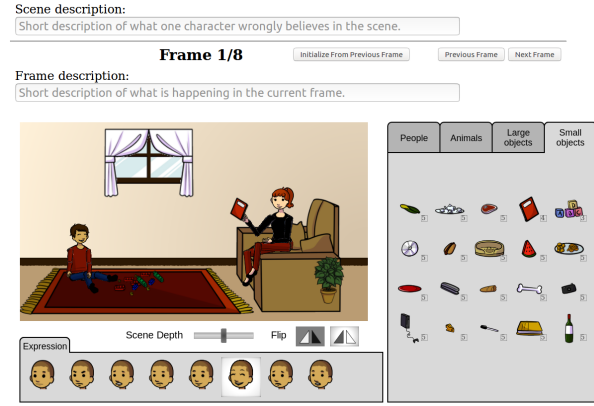


Figure 8: **Illustration interface:** This is the tool workers used to illustrate scenes. The people, animals, and objects available in the right pane were chosen randomly to diversify our dataset.

The scenes shown in the illustration quiz were chosen to highlight common mistakes we saw in a small pilot experiment we ran prior to collecting the main dataset. We found that adding the quiz significantly improved the quality of scenes in the main dataset as compared to the pilot experiment.

A.2. Illustration Interface

Figure 8 shows the illustration interface. There are four tabs to the right of the scene for choosing people, animals, large objects, and small objects to add to the scene. After illustrating a scene, workers also provided a scene-level description and eight frame-level descriptions. These descriptions were used to help workers annotate our dataset, but were not used to train our model.

B. Annotating the Dataset

In this section, we describe how workers annotated scenes in our dataset.

B.1. Annotation Quiz

Before workers could start annotating scenes, they completed a short quiz. In this quiz, we showed workers a couple scenes and accompanying questions. The workers were asked how each character in each scene would answer this question. Figure 10 shows two frames from two scenes in the quiz. Workers saw all 8 frames for each scene. In the frame on the left, the boy would answer “yes” because the boy knows he (the boy) did take the bike; the man would answer “no” because he thinks the boy did not take the bike.

B.2. Annotation Interface

After completing the annotation quiz, workers annotated scenes from our dataset. First, workers studied the scene,


Step 1 of 3: Watch the scene

In this task, you are going to watch a cartoon story about a character that incorrectly believes something. You can see your story below by pressing the Next/Previous buttons. Watch your story and try to figure out what the characters believe at each frame.

Most scenes should make sense. If this scene does not, check that checkbox below, and continue with the task anyways.

Scene Description: woman wonders where the steak went

Frame 4 of 8



Frame Description: she places the steak on the grill

☐ Scene does not make sense


Step 2: Frame 2 of 8



Scene Description: woman wonders where the steak went

Question:

Correct Answer: ☐ Yes ☐ No

 answers: ☐ Yes ☐ No


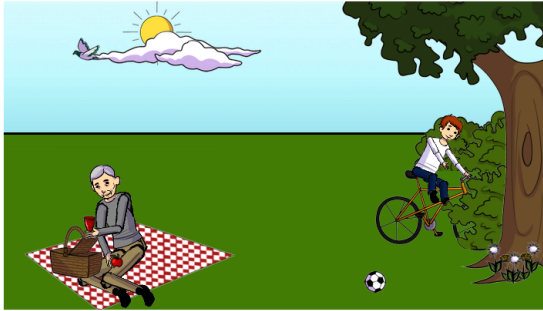
 answers: ☐ Yes ☐ No


Figure 9: **Annotation interface:** In the first part of the annotation step (**left**), workers studied the scene. In the second part (**right**), workers wrote questions and answers about each frame. These questions and answers were used to determine which characters were mistaken. The third step (not shown) allowed workers to submit feedback.

Frame 3



Question: Did the boy take the bike?

Correct answer: Yes

 answer: ☐ Yes ☐ No


 answer: ☐ Yes ☐ No

Figure 10: **Annotation quiz:** One example from the annotation quiz. Workers saw the entire 8-frame scene when answering questions about this frame.

as shown in Figure 9 (left). Second, workers wrote a question that some character would answer incorrectly in some frame, as shown in Figure 9 (right). Workers also pre-

dicted how each character would answer the question in each frame. Note that these questions and answers were only used to identify mistaken characters. They were not used to train our model.

We showed the annotators both the scene-level description and the frame-level descriptions. These helped annotators understand the problem we were studying. Importantly, the scenes were illustrated so it is possible to understand the scenes without reading these descriptions.

C. Animation

To provide another way of understanding our dataset, we animated the scenes. Because we have access to the generative parameters for each scene, it is easy to interpolate between frames. Note that the interpolated frames were not used to train our model. Rather, these videos highlight how access to the generative parameters are a unique strength of our dataset. These videos can be seen on the project webpage:

<http://people.csail.mit.edu/bce/mistaken/>.